CLAIMS

1. (Currently Amended) An interactive computer-implemented system for specifying and executing temporal order events, comprising a processor executing:

a constraint component that receives, from a user, loose temporal constraints associated with a plurality of events, wherein the loose temporal constraints specify information about execution of ~~an event comprising a start time or a stop time and event execution relative to other events~~ the plurality of events, and wherein the loose temporal constraints specify relative timing information but not exact literal times that that each of the plurality of events is to be executed;

a system information component that receives execution system information comprising one or more of available memory, cache coherency, data throughput or number of processors; and

an order component that ~~determines,~~ determines a plurality of event execution orders in accordance with the loose temporal constraints and via utility-based analysis of the execution system information, and selects an optimal event execution order from the plurality of event execution orders based on the execution system information; wherein:

each of the plurality of execution orders specifies exact literal times each of the plurality of events is executed;

the exact literal times are consistent with the loose temporal constraints; and

all of the plurality of execution orders do not provide the same specific temporal constraints on the plurality of events, but all of the plurality of execution orders are based on the loose temporal constraints.

2. (Currently Amended) The system of claim 1, wherein [[the]] a constraint from among the loose temporal constraints is an event start and/or a stop time.

3. (Currently Amended) The system of claim 1, wherein [[the]] a constraint from among the loose temporal constraints is event duration and/or a filter.

4. (Canceled)

5. (Currently Amended) The system of claim 1, ~~further comprising a~~ wherein the system information component ~~that~~ provides information about an execution system to the order component to facilitate selection of [[an]] the optimal event execution order.

6. (Canceled)

7. (Original) The system of claim 5, the information about an execution system includes data throughput rate.

8. (Currently Amended) An interactive computer-implemented system for specifying and executing temporal order events, comprising a processor executing:

a display component that provides a plurality of object workspaces, wherein the workspaces are user interfaces including a past, present and future space, wherein the present space is an editable area, and wherein the past and future space specify temporal constraints associated with a plurality of events; [[and]]

a design component that temporally associates and disassociate objects in the editable area, wherein the design component receives, from a user, loose temporal constraints governing event execution orders; and ~~area generating a plurality of event execution orders and determines an optimal execution order of events from the plurality of event execution orders based at least in part on the object associations specifying temporal constraints wherein non-associated objects order of execution is determined, via utility based analysis of an executing system information comprising available memory, cache coherency, data throughput and number of processors.~~

an order component that determines a plurality of event execution orders, wherein:

each of the plurality of execution orders provides a sequence by which the plurality of events could be executed in accordance with the loose temporal constraints;

each of the plurality of execution orders specifies exact literal times each of the plurality of events is executed;

the exact literal times are consistent with the loose temporal constraints;

all of the plurality of execution orders do not provide the same specific temporal constraints on the plurality of events; and

the order component selects an optimal event execution order from the plurality of event execution orders in accordance with execution system information.

9. (Previously Presented) The system of claim 8, object workspaces that facilitate a graphical-based approach to specify relationships amongst objects.

10. (Canceled)

11. (Original) The system of claim 8, non-associated objects are executed randomly.

12. (Original) The system of claim 8, the design component comprising a specification component that receives hard start and/or end times for events associated with objects.

13. (Original) The system of claim 8, the design component temporally associates objects as a function of respective location in the editable area.

14. (Original) The system of claim 8, further comprising a duration component that receives information regarding event duration.

15. (Original) The system of claim 8, the design component receives and executes information related to nested events associated with respective objects.

16. (Original) The system of claim 8, further comprising a policy component that applies predefined rules to execution of the objects.

17. (Original) The system of claim 8, further comprising a policy component that applies predefined rules to editing of the objects.

18. (Original) The system of claim 8, the design component receives and executes information regarding hierarchical relationship of respective objects.

19. (Original) The system of claim 8, the design component receives and executes information regarding dependency relationship of respective objects.

20. (Original) The system of claim 8, further comprising a query component that searches for events that satisfy a query, and displays objects associated with the events in temporal order.

21. (Previously Presented) The system of claim 20, the query component provides context information for respective objects.

22. (Previously Presented) The system of claim 8, objects placed in the past area are executed prior to objects in the present area and objects placed in the future area are executed after objects in the present area.

23. (Canceled)

24. (Original) The system of claim 8, the design component associates objects in a non-linear conditional manner.

25. (Original) The system of claim 8, the design component associates objects via iterative loops.

26. (Original) The system of claim 8, the design component associates objects based on a specified version.

27. (Currently Amended) A computer-implemented method for specifying and executing temporal order events comprising the following computer executable instructions stored on a tangible computer readable medium:

receiving, from a user, loose temporal constraints associated with a plurality of events, wherein the loose temporal constraints ~~specifying~~ specify information about execution of the plurality of events, and wherein the loose temporal constraints do not specify exact literal times that that each of the plurality of events is to be executed;

receiving execution system information comprising one or more of available memory, cache coherency, data throughput or number of processors;

generating a plurality of execution orders for the plurality of events in accordance with the loose temporal constraints, wherein:

each of the plurality of execution orders specifies exact literal times each of the plurality of events is executed;

the exact literal times are consistent with the loose temporal constraints; and

all of the plurality of execution orders do not provide the same specific temporal constraints on the plurality of events, but all of the plurality of execution orders are based on the loose temporal constraints and on the execution system information;

selecting an optimal event order based in part on the system execution information;

~~receiving execution system information comprising one or more of available memory, cache coherency, data throughput or number of processors; and~~

outputting [[an]] the optimal event execution order ~~selected from the plurality of execution orders based at least in part on execution system information~~.

28. (Canceled)

29. (Currently Amended) A method, comprising: ~~for object authoring implemented on a computer comprising the following computer executable instructions stored on a tangible computer readable medium:~~

storing, in a memory communicatively coupled to a processor, computer-executable instructions for performing the method, wherein the method orders events to be executed by an execution system;

executing the instructions on the processor;

according to the instructions being executed:

receiving object data associated with events from a workspace including at least one of a past, present, or future area;

associating objects temporally based at least in part upon relative object locations;

generating a plurality of execution orders based at least on the temporal association of the objects;

wherein the received object data is from a user, and comprises loose temporal constraints associated with a plurality of events, wherein the loose temporal constraints specify information about execution of the plurality of events, and wherein the loose temporal constraints do not specify exact literal times that that each of the plurality of events is to be executed;

receiving execution system information comprising one or more of available memory, cache coherency, data throughput or number of processors;

generating a plurality of execution orders for the plurality of events in accordance with the constraints, wherein:

each of the plurality of execution orders specifies exact literal times each of the plurality of events is executed;

the exact literal times are consistent with the loose temporal constraints; and

all of the plurality of execution orders do not provide the same specific temporal constraints on the plurality of events, but all of the plurality of execution orders are based on the loose temporal constraints;

selecting an execution order of events from the plurality of event execution orders based at least on information comprising available memory, cache coherency, data throughput and number of processors.

30. (Original) The method of claim 29, further comprising associating objects based on one or more operational objects.

31. (Previously Presented) The method of claim 30, wherein the operational objects correspond to at least a loop, a trigger, a conditional and hard start and/or stop times.

32-34. (Canceled)

35. (Original) The method of claim 29, wherein objects are associated in a non-linear conditional manner.

36. (Original) The method of claim 29, wherein the objects are associated via iterative loops.

37-38. (Canceled)

39. (Previously Presented) The system of claim 8, wherein information about event start and stop times and event duration is communicated with a particular object.

40. (Previously Presented) The system of claim 39, wherein the objects are bars and fuzzy edges on the bars indicate an unspecified time.

41. (Previously Presented) The system of claim 40, wherein the fuzzy edges on at beginning of the bar indicate an unspecified start time and the fuzzy logic on at end of the bar indicates an unspecified end time and/or duration.

42. (Previously Presented) The system of claim 40, wherein hard bold edges on the bar specifies specific start and/or stop time.

43. (Previously Presented) The system of claim 1, wherein the execution system information comprises at least available memory, cache coherency, data throughput and number of processors of the system.

44. (Previously Presented) The system of claim 8, wherein the temporal constraints comprising one specific event must finish before another specific event starts.

45. (Previously Presented) The system of claim 8, wherein the past and future space provide a context for navigation to a user during application development.